

REMARKS

Claims 54, 55, 67-72, 74, 75, 85-87, 89, 90, and 99-103 are cancelled. Claims 51, 53, 57, 63, 73, 80, 83, 88, and 95 are amended. New claims 104-106 are added. Support for the new claims is found at p. 15, lines 7-8. No new matter is added.

Claims 51-103 were subject to restriction as being distinct subcombinations from original claims 1-50. Applicants respectfully traverse this restriction.

First, the Examiner has miss-classified claims 1-50. Subclass 380/268¹ is indented under subclass 380/255, "Subject matter including cryptographic structure or methods which protect the transmission or reception of information." The methods and systems of the present invention do not protect any transmission or reception of any information. Rather, they protect (e.g., preserve the secrecy of) a secret key embedded in a processing module, such as the Subscriber Identity Module (SIM) card of a GSM mobile telephone. See Applicant's specification, p. 1, lines 5-15. In particular, the present invention is unrelated to pseudo-random sequence scrambling, the subject matter of subclass 380/268.

Cancelled claim 1 recited "scheduling said calculations using a precomputed, fixed randomization schedule in such a way that externally observable parameters of the device cannot be associated to particular pieces, bits, symbols or values of said secret information." There are two embodiments wherein calculations are scheduled using a precomputed, fixed randomization schedule to thwart external observation of the secret information. In both embodiments, the secret information is used in the same way, to perform the same calculation, which is briefly described:

Exponentiating a large number by a large secret key a fundamental mathematical operation in the cryptographic arts. Straightforward calculation of this mathematical operation, however, is not practical due its computational complexity. One well-known

¹ The Office Action state class 380/168, which does not exist.

method of reducing the computational complexity of this mathematical operation is by selectively calculating successive squares of the large number (base) if the corresponding bit of the secret key (exponent) is a one, multiplicatively accumulating the squares, and reducing the result modulo-N. Where the corresponding bit position of the key (exponent) is zero, the square is not calculated (and, obviously, not accumulated). See p. 3, line 3 – p. 5, line 5 for an explanation of the mathematical algorithms.

Implementing this algorithm exposes the secret key (the exponent) to one observing power consumption, timing, and other physical properties of the circuit performing the calculation. Obviously, a loop iteration that checks a key bit, finds a bit value of zero, and iterates to the next bit, will consume little power and will execute quickly. On the other hand, a loop iteration that checks a key bit, finds a bit value of one, and in response calculates a partial square, multiplicatively accumulates it, and reduces the result modulo-N, will consume more power and take longer to complete. This difference in either externally observable parameter may be exploited to ascertain the secret key bit sequence. The present invention hides the value of the secret key (exponent) from external observation during the above-described calculation in two ways.

As claimed in amended claims 51, 73 and 88, dummy calculations are performed in some of the “zero” secret key bit positions (which calculations would ordinarily be skipped). The results are not multiplicatively accumulated, but are simply discarded. The power and time required to perform the successive squares operation obscures the secret key from detection by observing external properties. Which of the nominally half of the zero bits will trigger a dummy calculation are determined by a precomputed, fixed randomization schedule in the form of an indicator word the same length as the secret key. A subset (e.g., a third) of the bits in the indicator word have a value of one, and these bits are distributed randomly. Where a one in the indicator word coincides with a

zero in the secret key, a dummy calculation is performed. Where a zero in the indicator word coincides with a zero in the secret key, no calculation is performed. For a one in a bit position in the secret key, the corresponding bit position in the indicator is a don't care – that is, the calculation will be performed and the result multiplicatively accumulated regardless of the indicator word bit value. See p. 12, line 12 – p. 14, line 8. In this embodiment, the order of calculations is not changed or randomized. The indicator word is a precomputed, fixed randomization schedule as recited in claim 1, that determines whether dummy calculations will or will not be inserted in zero bit positions of the secret key.

Another approach to thwarting the external observation of the secret key is claimed in claims 63, 80, and 94, as amended herein. In this embodiment, a group (e.g., eight) of successive squares is calculated unconditionally, and stored in a local memory. The corresponding bit positions of the secret key are then inspected, and if the bit is a one, the corresponding result is retrieved from memory and multiplicatively accumulated. For bit positions of the secret key having a zero value, the next successive square is calculated and stored in the corresponding memory location. See p. 14, line 9 – p. 16, line 38.

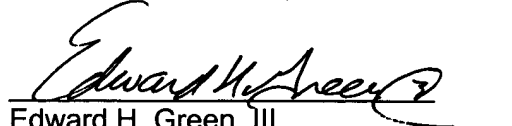
Optimizing this algorithm results in strings of successive square calculations, followed by varying occurrences of multiplication operations (from the multiplicative accumulations) and additional successive square calculations. See p. 16, line 39 – p. 18, line 30. As described at p. 18, line 32 – p. 19, line 19, if the externally observable parameters can be distinguished between an instruction that squares a number and an instruction that multiplies it by itself in a conventional multiply operation (which is plausible, as the square is a special case that is particularly computationally efficient), some bit positions of the secret key may be determined by counting successive square operations between multiply operations. To hide this information, whenever two

successive square operations are to be performed, either the first or second may randomly be chosen and implemented as a multiply of a number by itself, rather than a square (for a string of three square's, the middle one will be converted to a multiply). Since the secret key is known and long-term fixed, the order of operations that will be performed on any base number can be determined. Thus the random selection of the first or second of a pair of successive square operations, to be implemented as a multiply, may be predetermined and stored. This information comprises a precomputed, fixed randomization schedule, as recited in claim 1. Note that only the implementation of a mathematical operation is changed according to the precomputed, fixed randomization schedule; the order of calculation of the successive squares does not change, and is not random.

Accordingly, both the embodiment of the present invention claimed in claims 51, 73 and 88, and that of claims 63, 80, and 94, schedule calculations using a precomputed, fixed randomization schedule in such a way that externally observable parameters of a device cannot be associated to particular pieces, bits, symbols or values of said secret information, as recited in claim 1. As such, the claims are not properly classified in separate classes, and restriction is improper.

The Examiner is respectfully directed to Applicants' Response filed September 17, 2004 for arguments distinguishing the claims from the prior art cited in the § 102 and § 103 rejections imposed in the Office Action of June 15, 2004.

Respectfully submitted,
COATS & BENNETT, P.L.L.C.



Edward H. Green, III
Attorney for Applicants
Registration No.: 42,604
P.O. Box 5
Raleigh, NC 27602
Telephone: (919) 854-1844

Dated: March 31, 2004